

Amendments to the Specification

Please replace the Paragraph beginning on page 3, line 16, with the following:

Generally, the computer 102 operates under control of an operating system 108 stored in the memory 106, and interfaces with the user to accept inputs and commands and to present results through a graphical user interface (GUI) module 118A. Although the GUI module 118A is depicted as a separate module, the instructions performing the GUI functions can be resident or distributed in the operating system 108, the ~~computer~~ application program 110, or implemented with special purpose memory and processors. The computer 102 also implements a compiler 112 which allows [[an]] the application program 110 written in a programming language such as COBOL, C++, FORTRAN, or other language to be translated into processor 104 readable code. After completion, the application program 110 accesses and manipulates data stored in the memory 106 of the computer 102 using the relationships and logic that was generated using the compiler 112. The computer 102 also optionally comprises an external communication device 130 such as a modem, satellite link, Ethernet card, or other device for communicating with other computers.

Please replace the Paragraph beginning on page 3, line 29, with the following:

In one embodiment, instructions implementing the operating system 108, the ~~computer~~ application program 110, and the compiler 112 are tangibly embodied in a computer readable medium, e.g., data storage device 120, which could include one or more fixed or removable data storage devices, such as a zip drive, floppy disc drive 124, hard drive, CD-ROM drive, tape drive, etc. Further, the operating system 108 and the ~~computer~~ application program 110 are comprised of instructions which, when read and executed by the computer 102, causes the computer 102 to perform the steps necessary to implement and/or use the present invention. ~~Computer~~ Application program 110 and/or operating instructions may also be tangibly embodied in memory 106 and/or data communications device(s) 130, thereby making a computer program product or article of manufacture according to the invention. As such, the terms "article of manufacture," "program storage device" and "computer program product" as used herein are intended to encompass a computer program accessible from any computer readable device or media.

Please replace the Paragraph beginning on page 5, line 10, with the following:

Another possible solution to this problem is to separately download source data from the client computer 102 to the printer server 204, and then provide some options to modify the source data at the printer server 204. Unfortunately, this approach requires a very large investment on the part of the printer 206 vendor to provide just a subset of impositioning capability. Performance issues from the printer server 204 to the printer 206 are not addressed, nor is version control, and neither of the foregoing alternatives is suitable for integration with variable data applications.

Please replace the Paragraph beginning on page 5, line 23, with the following:

The print optimizer 302 includes a resource checking module 320 for determining if a resource provided by the data generator 310 is already associated with an identifier and stored in the ~~print~~ printer server 306 and/or a memory 326 of the printer 308. The print optimizer 302 also includes a processing module 322 which containerizes or RIPs (rasterize and print) resources in the impositioned document to a printer-renderable form. The print optimizer 302 also maintains and manages a database 324 which associates resource identifiers with AFP identifiers such as the resource name and object ID.

Please replace the Paragraph beginning on page 8, line 6, with the following:

FIG. 5B is a block diagram further illustrating the operations performed in customizing the impositioned document (e.g. those performed in block 410 of FIG. 4). If no job ticket has been created, source data is loaded into the ~~data~~ document formatter 316 of the impositioning module 304, and job ticket identifiers are associated with the variable data tags in the impositioned document. The variable data tags identify where variable data is to be inserted. This is depicted in blocks 508A and 510. If a job ticket has already been created, the job ticket can be imported into the ~~data~~ document formatter 316, as shown in block 508B. Further, if the only change in the impositioned document is a change in variable data, the customization database can be updated, as shown in block 508C. The variable data tags are then replaced with customization data or references to customization data, as shown in blocks 512A and 512B. In one embodiment, this is accomplished with the use of the customization database 318 which has one or more keys 344 which index columns of data to be inserted at desired places in the document. For example, a customer ID can be used to retrieve a name and address for a personalized form letter.

Please replace the Paragraph beginning on page 9, line 5, with the following:

Non-printer captureable resources (resources that are not supported by the printer ~~[[206]]~~ 308) are converted to an image (using, for example, a raster image processor, or RIP in the printer ~~server~~ 308) or a command set for printing an image (such as the FS10 command set used in conjunction with the image object content architecture (IOCA)). The image is then associated with a resource name (e.g. Resource C in FIG. 3). These operations are shown in blocks 526 and 528.

Please replace the Paragraph beginning on page 9, line 11, with the following:

The processed resources are then stored in the ~~print~~ printer server 306. The database 324 in the print optimizer 302 is then updated to correlate the first identifier from the job ticket (e.g., the file name of the document and page number) and the second identifier (AFP identifier or MODCA ID), as shown in block 532. Thereafter, the print optimizer ~~[[534]]~~ 302 responds to database queries from the job ticket transformer 330 in the printer server 306 to permit translation of first (job ticket) identifiers to second (AFP or MODCA) identifiers, as shown in block 534.

Please replace the Paragraph beginning on page 10, line 13, with the following:

In one embodiment, the collateral information is designated as an in-line resource for the printfile and included with IOBs. Such resources can remain in the source format. The resulting AFP data stream is simple and small, thus simplifying the development of AFP drivers, minimizing the processing time required to create an AFP printfile, and minimizing network traffic from the ~~client~~ computer 102 to the printer server 306.

Please replace the Paragraph beginning on page 10, line 24, with the following:

The user may provide commands to modify the presentation of ~~[[a]]~~ the document or any resource within the document. A user interface is provided to allow the user to specify the desired page locations, rotations, and scaling for each object on each signature. The user can also specify collateral information such as cut marks and spine text. The user commands are accepted in the imposition engine ~~[[302]]~~ 314. A new skeleton document 602 is created in accordance with the user command(s), and the document is rendered using the modified skeleton document and the loaded resources.